Information Networks: PageRank

Web Science (VU) (706.716)

Elisabeth Lex

ISDS, TU Graz

June 18, 2018

Repetition

- Information Networks
- Shape of the Web
- Hubs and Authorities

PageRank

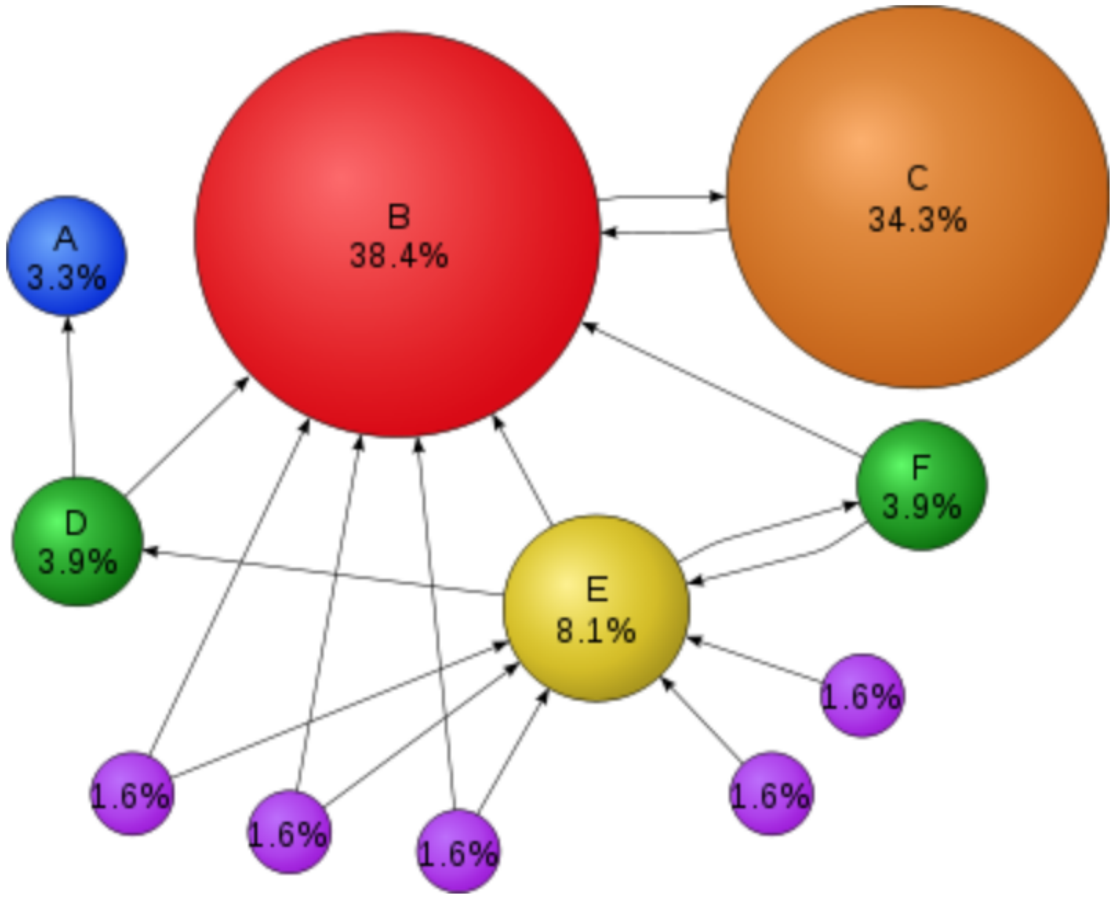
PageRank

- Intuition from last week: links as votes (HITS algorithm)
- Page more important if it has many in-links
- Do you think that all in-links are equal?

PageRank

- Intuition from last week: links as votes (HITS algorithm)
- Page more important if it has many in-links
- Do you think that all in-links are equal?
- No! Links from important pages are more important

Example for PageRank Scores



PageRank vs HITS

- Can you think of the major difference between PageRank and HITS?

PageRank vs HITS

- Can you think of the major difference between PageRank and HITS?
- Unlike HITS, PageRank is independent of the search query!

PageRank Algorithm in a Nutshell

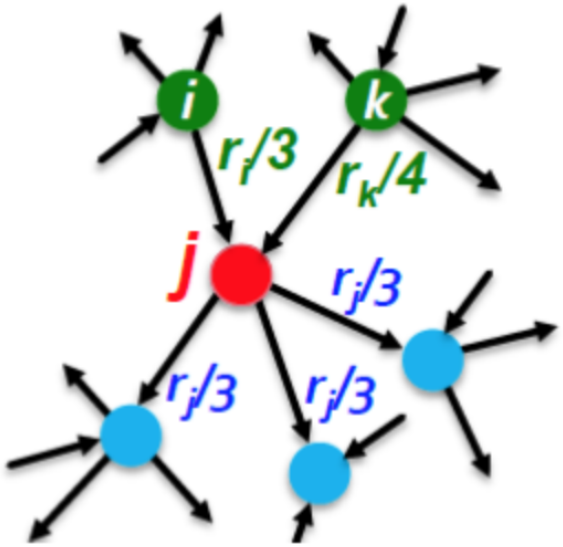
- Start with a set of pages
- Crawl the Web to determine link structure
- Assign each page an initial rank of $1/N$
- Successively update rank of each page by adding up weight of each page that links to it divided by nr of out-links of the referring page

PageRank: Basic Formulation

- PageRank calculation also starts with simple voting based on in-links and gets then repeatedly improved
- A link's vote is proportional to importance of its source page
- Let page j have importance r_j and n out-links
- Then, each link gets r_j/n votes
- Importance of page j: sum of the votes on its in-links

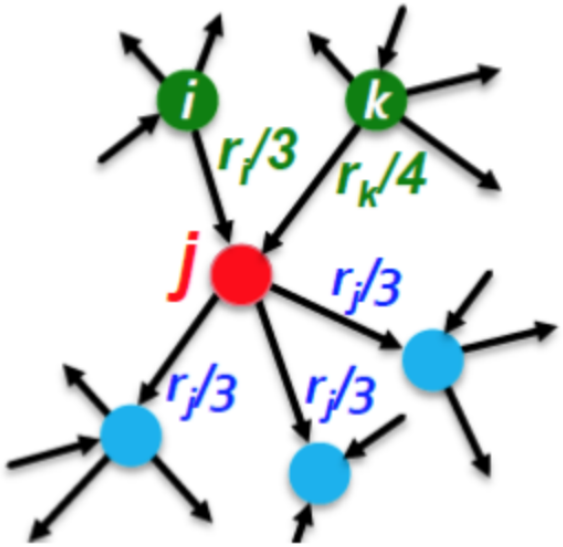
PageRank: Basic Formulation

- PageRank calculation also starts with simple voting based on in-links and gets then repeatedly improved
- A link's vote is proportional to importance of its source page
- Let page j have importance r_j and n out-links
- Then, each link gets r_j/n votes
- Importance of page j: sum of the votes on its in-links



PageRank: Basic Formulation

- PageRank calculation also starts with simple voting based on in-links and gets then repeatedly improved
- A link's vote is proportional to importance of its source page
- Let page j have importance r_j and n out-links
- Then, each link gets r_j/n votes
- Importance of page j: sum of the votes on its in-links



$$r_j = r_i/3 + r_k/4$$

The "Flow" model of PageRank

- We learned that a page is important if it is linked from other important pages
- Plus, a "vote" (via in-link) from an important page is worth more
- Based on that, we can define a "rank" r_j for a page j

$$r_j = \sum_{i \to j} \frac{r_i}{d_i} \tag{1}$$

where d_i is the out-degree of node i

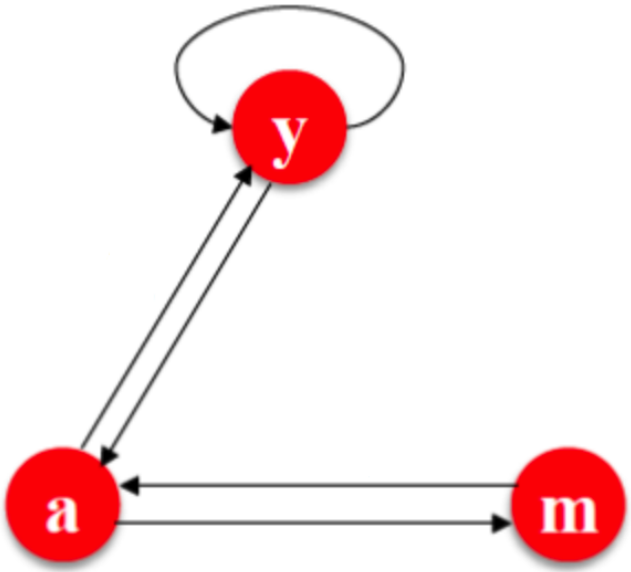
The "Flow" model of PageRank

- We learned that a page is important if it is linked from other important pages
- Plus, a "vote" (via in-link) from an important page is worth more
- Based on that, we can define a "rank" r_j for a page j

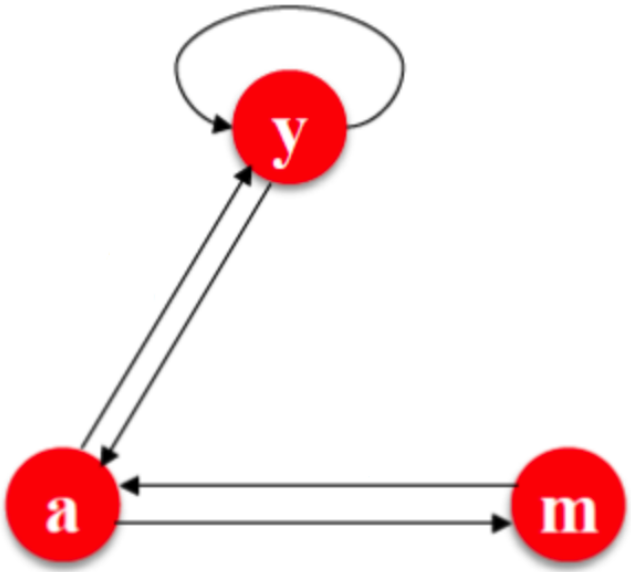
$$r_j = \sum_{i \to j} \frac{r_i}{d_i} \tag{1}$$

  where d_i is the out-degree of node i
- Intuitively, we can think of PageRank as a kind of fluid that "flows" through the network
- The fluid passes from node to node across links
- It pools at the nodes that are the most important

Example

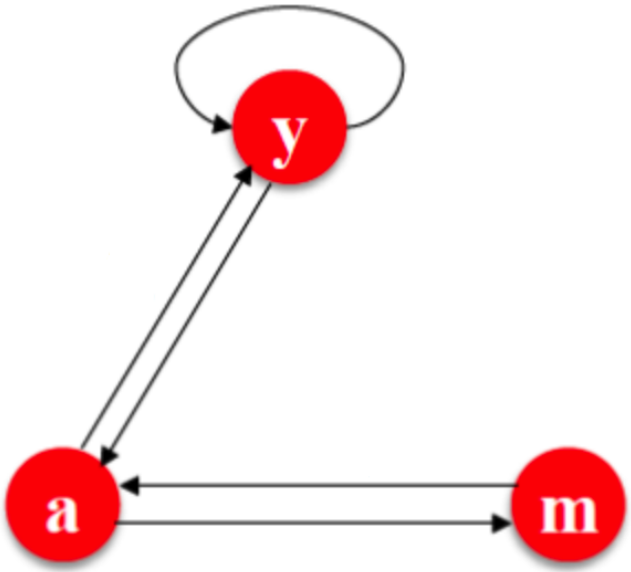


Example



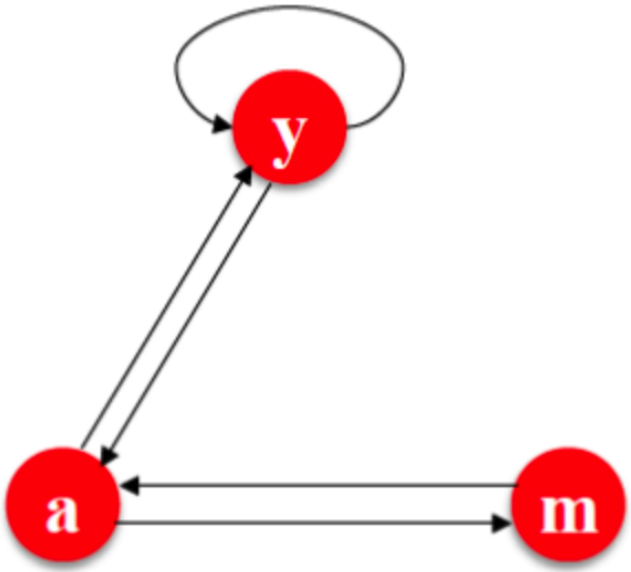
$$r_y = r_y/2 + r_a/2$$

Example



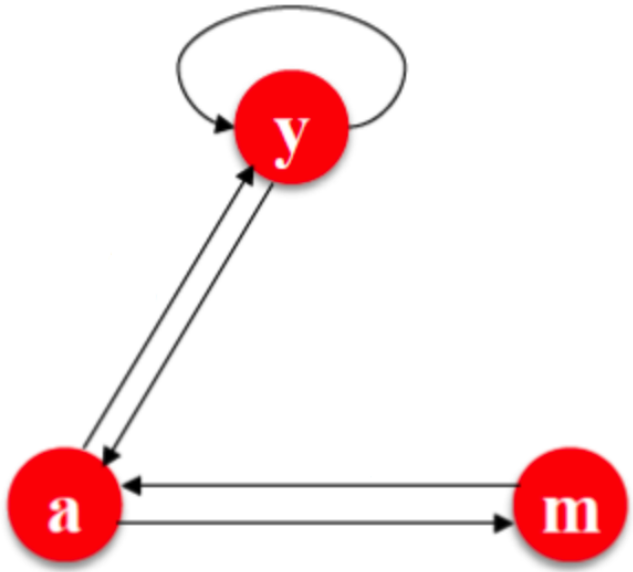
$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$

Example



$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

Example



$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

These are called "flow" equations

Solving the equations

- In the last example: 3 equations, 3 unknowns, no constants
- This means, there is no unique solution to them
- We need an additional constraint to enforce unique solution
- Ranks need to sum up to 1, i.e.
- This means that for our small graph:
  $r_y = r_y/2 + r_a/2$
  $r_a = r_y/2 + r_m/1$
  $r_m = r_a/2$
  $r_y + r_a + r_m = 1$
- So we have 3 unknowns and 4 equations - solvable through elimination
- Solution: $r_y = 2/5$, $r_a = 2/4$, $r_m = 1/5$

Matrix Formulation

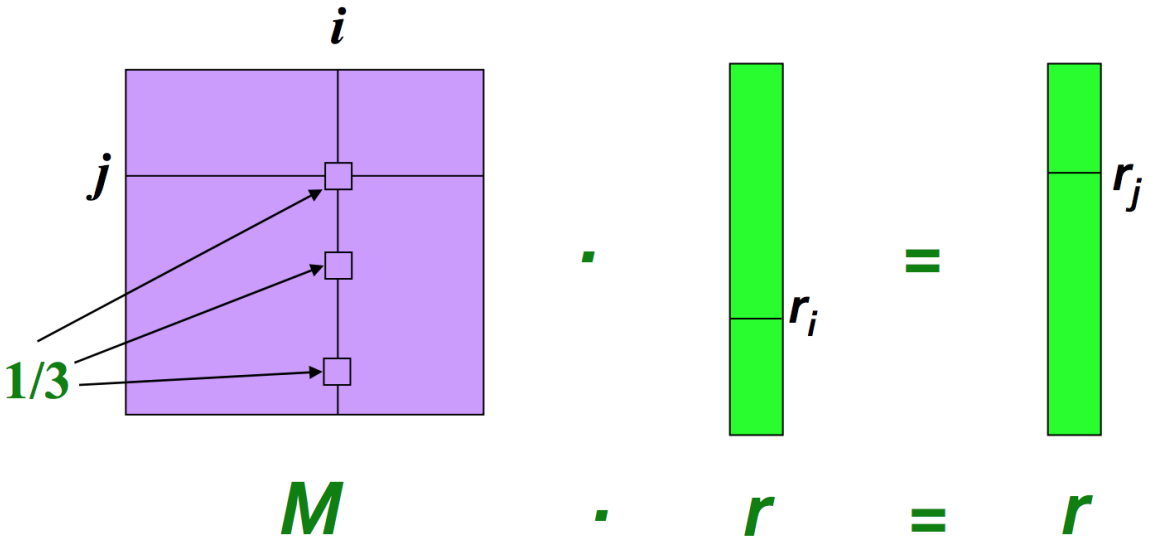
- Elimination does not apply to large scale graphs
- We need a different formulation of the problem
- Matrix formulation: Stochastic adjacency matrix M
  - Let page i have d_i out-links
  - if $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$ where M is a column stochastic matrix, i.e., columns sum up to 1
- Rank vector r: vector with an entry per page
- Length of r is the number of pages in our sample
  - r_i corresponds to pagerank score of page i
  - $\sum_i r_i = 1$ due to constraint of flow equations
- Thus, we can write the flow equations from before as vector-matrix product:
  $r = M \cdot r$

Matrix Formulation: Example

- Flow equation as sum: $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
- Flow equations in matrix form as vector-matrix product: $r = M \cdot r$
- Let's assume page i, which has links to 3 pages, i.e. $d_i = 3$. One of the pages it links to is page j.

Matrix Formulation: Example

- Flow equation as sum: $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
- Flow equations in matrix form as vector-matrix product: $r = M \cdot r$
- Let's assume page i, which has links to 3 pages, i.e. $d_i = 3$. One of the pages it links to is page j.



Matrix Formulation

- Recursive maxtrix equation $r = M \cdot r$ resembles an eigenvalue problem
- Eigenvalue problem definition:

Definition

Vector x is an eigenvector with the corresponding eigenvalue λ if they are a solution to the following problem: $Ax = \lambda x$

Note that A is given, and we aim to compute x and λ

Matrix Formulation

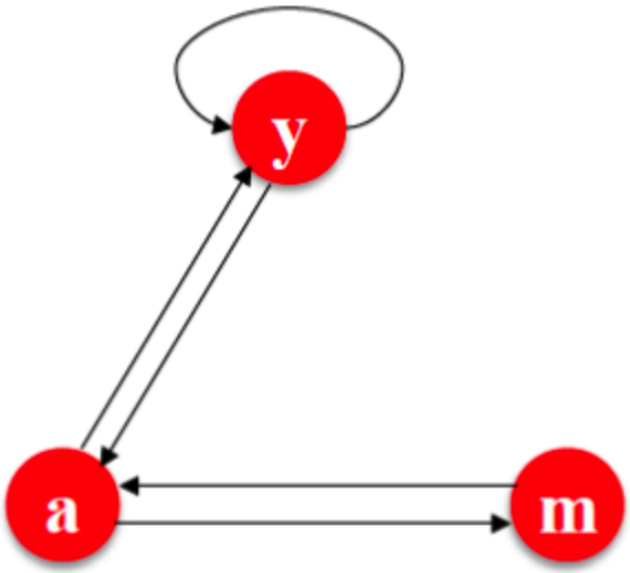
- Equation $r = M \cdot r$ looks similar to $Ax = \lambda x$
- In other words: rank vector r is an eigenvector of stochastic web matrix M
- What is the value of λ?

Matrix Formulation

- Equation $r = M \cdot r$ looks similar to $Ax = \lambda x$
- In other words: rank vector r is an eigenvector of stochastic web matrix M
- What is the value of λ?
- Rank vector r is not any eigenvector, but its *principal* eigenvector, i.e., its corresponding eigenvalue is 1, ergo $\lambda = 1$
- Reason: vector r has unit length (its coordinates are nonnegative and sum to 1, also called "stochastic vector")
- Plus, each column of M sums up to 1 (M is "column stochastic")
- This means: $M \cdot r \leq 1$
- Hence, largest eigenvalue of $M = 1$

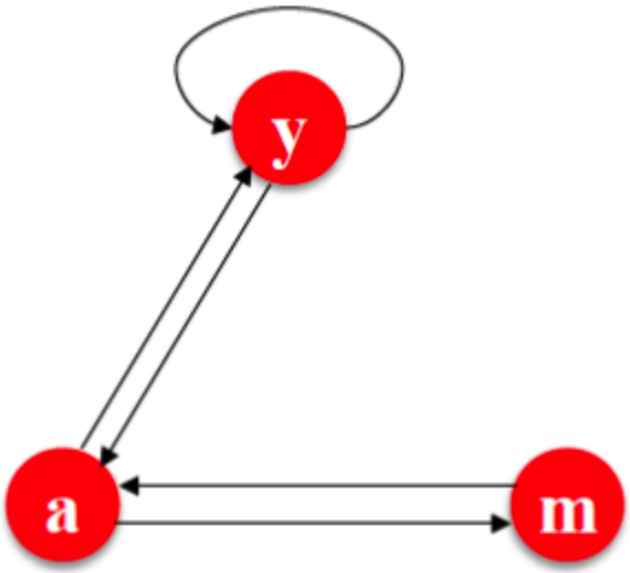
This can be efficiently solved for r using Power iteration method

Example:



$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

Example:

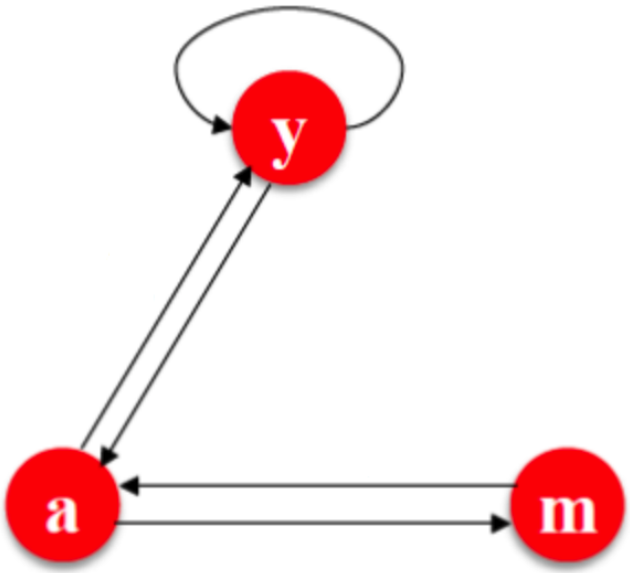


	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$r = M \cdot r$

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

Example:



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r = M \cdot r$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

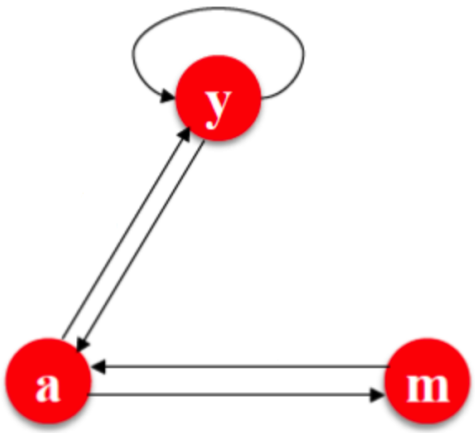
Power Iteration

- We assume N web pages
- Initialize $r^{(0)} = [1/N, ..., 1/N]^T$
- Iterate $r^{(t+1)} = M \cdot r^{(t)}$
- Stop when $|r^{(t+1)} - r^{(t)}|_1 < \epsilon$
- Algorithm:
  - We set r_j to $1/N$
  - First step: $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$.
  - Second step: $r = r'$. Go to first step until convergence.

Power Iteration: Example

Power Iteration: Example

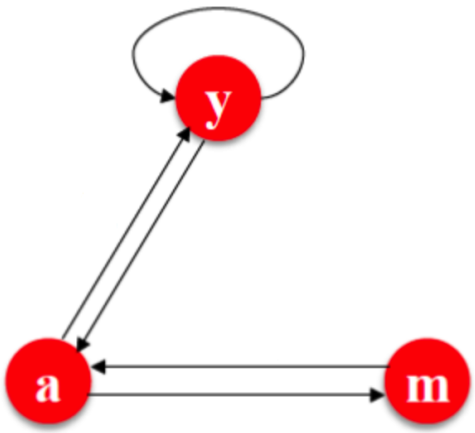
Example:



$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

Power Iteration: Example

Example:



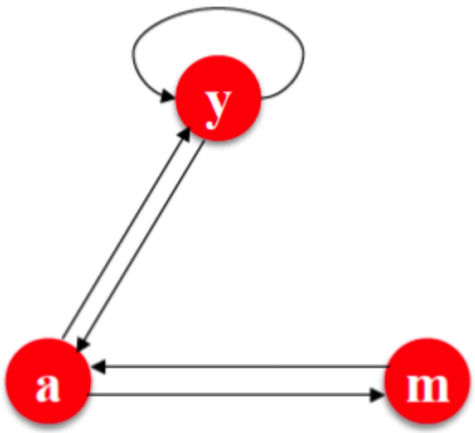
	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$r = M \cdot r$

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

Power Iteration: Example

Example:



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r = M \cdot r$$

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m/1$$
$$r_m = r_a/2$$

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2, …

Random Walks Interpretation: An equivalent definition of PageRank

- So far, we computed PageRank using flow equations and in terms of matrix formulation
- Now, we will look at an iterpretation what PageRank scores reflect
- Random Walk Interpretation
- PageRank scores equivalent to probability distribution of a random walker on the graph

Random Walk Interpretation: An equivalent definition of PageRank

Consider someone who is randomly browsing a network of Web pages - a "random web surfer"

- Surfer starts at any time t by choosing a page i at random, picking each page with equal probability
- At time $t+1$, surfer picks uniformly at random an out-going link from page i and follows it
- Ends up on some page j linked from i
- Process repeats indefinitely
- If page j has no out-going links, surfer stays
- This is called a **Random Walk** on the network

Random Walk Interpretation

With what probability is a random walker at time t at a given page?

- Let $p(t)$ be a vector whose coordinate i denote the probability that the surfer is at page i at time t
- Thus, $p(t)$ gives us a probability distribution over pages

Random Walk Interpretation

Where is the random walker going to be at time $t+1$?

- Random walker follows an out-going link uniformly at random
- Thus: $p(t+1) = M \cdot p(t)$
- Suppose random walk reaches a state $p(t+1) = M \cdot p(t) = p(t)$, then $p(t)$ is called stationary distribution of a random walk
- Remember: rank vector $r = M \cdot p(t)$
- In other words: r is a stationary distribution for the random walk

What does that mean?

- PageRank scores correspond to probability is at a given node at a given time step
- A side note: random walks are effectively Markov processes
- Why is that important? Because for graphs that satisfy certain conditions, the stationary distribution is unique and will be reached at some point regardless of the initial probability distribution at time $t = 0$
- This means that there are conditions under which PageRank vector r is unique and will be achieved regardless of initialization

Problems with real web graphs

- Spider traps: a group of one or more pages that have no links out of the group will eventually accumulate all the importance of the Web
  - What is the problem with that?

Problems with real web graphs

- Spider traps: a group of one or more pages that have no links out of the group will eventually accumulate all the importance of the Web
  - What is the problem with that?
  - Eventually, the group absorbes all the PageRank scores

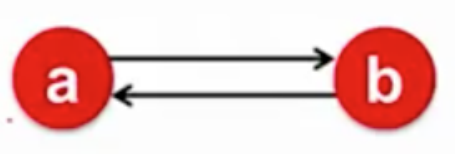
Problems with real web graphs

- Spider traps: a group of one or more pages that have no links out of the group will eventually accumulate all the importance of the Web
  - What is the problem with that?
  - Eventually, the group absorbes all the PageRank scores
- Dead ends: some pages have no out-links
  - What is the problem with that?

Problems with real web graphs

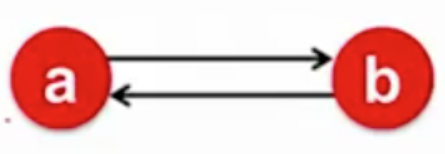
- Spider traps: a group of one or more pages that have no links out of the group will eventually accumulate all the importance of the Web
  - What is the problem with that?
  - Eventually, the group absorbes all the PageRank scores
- Dead ends: some pages have no out-links
  - What is the problem with that?
  - Dead ends make PageRank "leak out"

Spider Traps: Example



- We run the power iteration
- Remember: PageRank vector r is unique and stationary distribution will always be reached regardless of how we initialize
- What happens if we initialize $a = 1$ and $b = 0$?

Spider Traps: Example

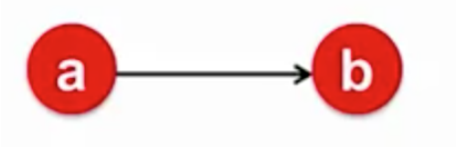


- We run the power iteration
- Remember: PageRank vector r is unique and stationary distribution will always be reached regardless of how we initialize
- What happens if we initialize $a = 1$ and $b = 0$?
- Both flip all the time, scores get passed to a and b and vice versa
- Power iteration does not converge - spider trap problem

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

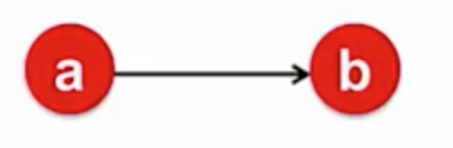
Iteration 0, 1, 2, ...

Dead end: Example



- We run the power iteration
- Remember: PageRank vector r is unique and stationary distribution will always be reached regardless of how we initialize
- What happens if we initialize $a = 1$ and $b = 0$?

Dead end: Example



- We run the power iteration
- Remember: PageRank vector r is unique and stationary distribution will always be reached regardless of how we initialize
- What happens if we initialize $a = 1$ and $b = 0$?
- In the first multiplication with matrix M, scores are flipped
- In the second step, the score 1 gets lost, as b can't pass it on
- This is also called "leaking out"

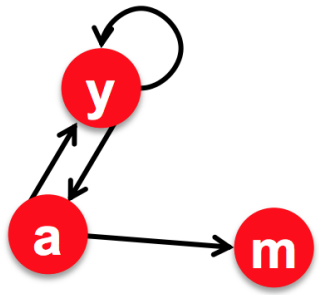
$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2, ...

Random Teleports as solution for spider traps

- At each step, random surfer has 2 options:
  - With probability β, follow a link at random
  - With probability $1 - \beta$, jump to some random page
  - In practice, $\beta = 0.8, 0.9$
- This enables random walker to teleport out of spider trap within few time steps

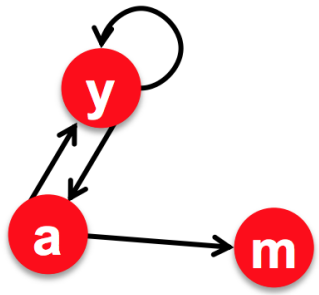
What about dead ends?



- Problem: Pages with 0 out-degree - their PageRank does not get distributed ("leaks out")
- What is apparent if we look at matrix M?

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

What about dead ends?

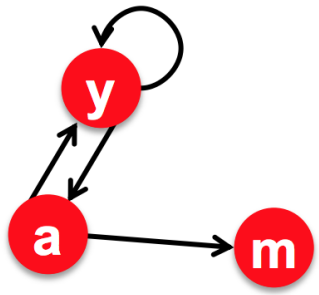


- Problem: Pages with 0 out-degree - their PageRank does not get distributed ("leaks out")
- What is apparent if we look at matrix M?

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

It is not stochastic anymore! Why?

What about dead ends?



- Problem: Pages with 0 out-degree - their PageRank does not get distributed ("leaks out")
- What is apparent if we look at matrix M?

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

It is not stochastic anymore! Why? Node m has 0 out-degree

Solution: Always teleport

- Power iteration: all vectors converge to zero
- Solution: Always teleport
- Follow random teleport links with probability 1 from dead-ends
- Update matrix M:

Solution: Always teleport

- Power iteration: all vectors converge to zero
- Solution: Always teleport
- Follow random teleport links with probability 1 from dead-ends
- Update matrix M:

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

- Why this update?

Solution: Always teleport

- Power iteration: all vectors converge to zero
- Solution: Always teleport
- Follow random teleport links with probability 1 from dead-ends
- Update matrix M:

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

- Why this update? In our example graph with 3 nodes, the random walker teleports out and with probability 1/3 lands on any other node in the network

Why do teleports help?

- Teleports help us make matrix M **stochastic** as we have seen that wrt dead ends
  - Whenever all the entries for a column in M are 0, we can replace them with $1/d_i$ where d_i is the out-degree of node i
- Teleports help us make matrix M **aperiodic**
  - Random walker can teleport out of loops
- Teleports help us make matrix M **irreducible**[1]
  - Teleports help us add random jumps to matrix M

[1] Irreducability: from any state, there is a non-zero probability of going from any one state to any other

Google's Solution: Random Jumps

Idea: combines all of this:

- Make matrix M stochastic, aperiodic, irreducible
- At each step, random surfer has 2 options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some random page
- PageRank equation [Brin-Page, 1998]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n} \tag{2}$$

Google Matrix

- PageRank equation:

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n} \tag{3}$$

- Google Matrix A:

$$A = \beta M + (1 - \beta)\frac{1}{n} e \cdot c^T \tag{4}$$

  where e is a vector of all 1s
- A is stochastic, aperiodic, irreducible, i.e. $r^{(t+1)} = A \cdot r^{(t)}$
- What is a good value for β? In practice, set to $\beta = 0.8, 0.9$, which means making 5 steps and jump
- What would $\beta = 0$ mean?

Google Matrix

- PageRank equation:

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n} \tag{3}$$

- Google Matrix A:

$$A = \beta M + (1 - \beta)\frac{1}{n} e \cdot c^T \tag{4}$$

  where e is a vector of all 1s
- A is stochastic, aperiodic, irreducible, i.e. $r^{(t+1)} = A \cdot r^{(t)}$
- What is a good value for β? In practice, set to $\beta = 0.8, 0.9$, which means making 5 steps and jump
- What would $\beta = 0$ mean? Random walker would jump all the time, all nodes in the network would have same PageRank

Google Matrix

- PageRank equation:

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n} \tag{3}$$

- Google Matrix A:

$$A = \beta M + (1 - \beta)\frac{1}{n} e \cdot c^T \tag{4}$$

  where e is a vector of all 1s
- A is stochastic, aperiodic, irreducible, i.e. $r^{(t+1)} = A \cdot r^{(t)}$
- What is a good value for β? In practice, set to $\beta = 0.8, 0.9$, which means making 5 steps and jump
- What would $\beta = 0$ mean? Random walker would jump all the time, all nodes in the network would have same PageRank
- And $\beta = 1$?

Google Matrix

- PageRank equation:

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n} \tag{3}$$

- Google Matrix A:

$$A = \beta M + (1 - \beta)\frac{1}{n} e \cdot c^T \tag{4}$$

  where e is a vector of all 1s
- A is stochastic, aperiodic, irreducible, i.e. $r^{(t+1)} = A \cdot r^{(t)}$
- What is a good value for β? In practice, set to $\beta = 0.8, 0.9$, which means making 5 steps and jump
- What would $\beta = 0$ mean? Random walker would jump all the time, all nodes in the network would have same PageRank
- And $\beta = 1$? No random jumps

A few words on actual computation

- Computing PageRank involves expensive matrix-vector multiplication
- Matrix A has no non-zeros, i.e. is a dense matrix, huge in terms of the Web
- Idea: Rearrange PageRank equation so it features matrix M (which is sparse) and not A

Limitations of PageRank

- PageRank measures general popularity / importance of a page - Why a problem?

Limitations of PageRank

- PageRank measures general popularity / importance of a page - Why a problem?
  - Neglects topic-specific authorities
  - Topic-specific PageRank
- Susceptible to Link spam
  - Link structures created to boost PageRank
  - Solution: TrustRank

Topic-Specific PageRank

- Intuition: Let the random surfer teleport to a random page that is chosen non-uniformly
- This helps us derive PageRank values that are tailored to meet specific topic needs
- E.g. a soccer fan might want pages on sports ranked higher
- So, random surfer should teleport to a random page that is about sports
- Caveat: Requires a collection of pages that are categorized as sports

Some Practical Examples for PageRank

- PageRank on protein interaction graphs[2]
- Social Media Analysis[3]
- Altmetrics and Analysis of Readership Data on Mendeley[4]

[2] http://rsos.royalsocietypublishing.org/content/2/4/140252.abstract

[3] http://www.cs.columbia.edu/~ecj2122/research/social_higgs/jubb_facheris_discovery_of_the_higgs.pdf

[4] http://arxiv.org/abs/1504.07482

Summary

We have learned about:

- PageRank: the Flow model, matrix formulation
- Random Walks
- Problems with PageRank: spider traps, dead ends
- Teleportation and random jumps as solution
- Some applications beyond web search and ranking

Thanks for your attention - Questions?

elisabeth.lex@tugraz.at

Slides use figures and content from Mining of Massive Datasets by Jure Leskovec, Anand Rajaraman, Jeff Ullman. See http://www.mmds.org/